ANH >

dongdu- A Vietnamese Word Segmentation tool

目次

Cấu trúc

```
dongdu
     |
     |---liblinear
     |     |
     |     |---(lược bỏ)
     |
     |---data
     |     |
     |     |---VNsyl.txt
     |     |
     |     |---wordlist.txt
     |
     |---configure.h
     |
     |---DicMap.h, DicMap.cpp
     |
     |---Feats.h, Feats.cpp
     |
     |---FeaturesSeclection.h, FeaturesSelection.cpp
     |
     |---Machine.h, Machine.cpp
     |
     |---StrMap.h, StrMap.cpp
     |
     |---SylMap.h, SylMap.cpp
     |
     |---predictor.cpp
     |
     |---learner.cpp
```

Ở đây, ý nghĩa của các file và folder:

- thư mục liblinear : thư viện máy học, thực hiện việc học máy.
- thư mục data : chứa dữ liệu. VNsyl.txt là file chứa các âm tiết tiếng Việt. wordlist.txt là file chứa các từ tiếng Việt.
- configure.h : file chứa những định nghĩa chung như các const hay typedef.
- predictor.cpp : thực hiện việc đánh giá và tách từ.
- learner.cpp : thực hiện việc phân tích dữ liệu đầu vào và học máy.

- các file .h và .cpp còn lại là các class sử dụng trong chương trình. Ý nghĩa và các thao tác cụ thể được trình bày ở dưới.

API

Để sử dụng DongDu như 1 phần của 1 chương trình khác, bạn cần những thư mục và file sau :

1. thư muc liblinear
2. thư mục data
3. các file :
   - configure.h
   - DicMap.h, DicMap.cpp
   - Feats.h, Feats.cpp
   - FeaturesSelection.h, FeaturesSelection.cpp
   - Machine.h, Machine.cpp
   - StrMap.h, StrMap.cpp
   - SylMap.h, SylMap.cpp
4. trong chương trình của bạn, bạn cần thêm
   ```
   #include "Machine.h"
   #include "configure.h"
   ```
5. Tạo 1 class mới
   - predictor - để tách một câu (hệ đánh giá):
     ```
     Machine predictor(3, "", PREDICT);
     predictor.load();
     ```
     - 3 là độ dài của W. Về ý nghĩa của W, xin đọc thêm tài liệu giải thích về thuật toán.
     - "" - giá trị không cần thiết
     - PREDICT : giá trị lựa chọn hệ, đánh giá hoặc học máy
   - learner - để tiến hành học máy (hệ học máy):
     ```
     Machine learner(3, "corpus_link", LEARN);
     ```
     - 3 : độ dài của W. Có thể tăng hay giảm tùy theo mục đích. Nếu tăng lên thì thường kết quả cũng tốt hơn, nhưng file dữ liệu và tốc độ xử lý sẽ chậm đi.
     - "corpus_link" : link đến file chứa dữ liệu huấn luyện.
     - LEARN : giá trị lựa chọn hệ.
6. Ở hệ PREDICT, class cung cấp thao tác `predictor.segment(string_)` để tách câu string_. Kết quả trả lại cũng là 1 string.

Class

DicMap

Class quản lý từ điển. DicMap nhập dữ liệu vào từ file wordlist.txt, và thực hiện những thao tác như kiểm tra xem 1 từ có trong từ điển hay không.

```
private :
    map<string, int> dmap_;
```

DicMap

mục đích	khởi tạo. Đọc từ file từ điển trong dữ liệu.
tham số	void
trả lại	void

isWord

mục đích	kiểm tra 1 từ có nằm trong từ điển hay không
tham số	string
trả lại	bool

StrMap

Class quản lý stt của các đặc trưng. Trong liblinear, các đặc trưng phải được đánh stt từ 1 trở đi, và phải liên tiếp nhau theo thứ tự xuất hiện.

StrMap có 2 hệ : LEARN hoặc PREDICT. Giữa 2 hệ này có 1 số khác biệt nhỏ. Vài class tiếp theo cũng có 2 hệ này.

```
    private:
        map<string, int> smap_;
        int size_;
        bool isfull_;
```

getNum

mục đích	lấy ra stt của 1 đặc trưng. Nếu đặc trưng này không có trong map, thì ở hệ LEARN, sẽ thêm đặc trưng này vào, và tăng thêm stt, còn ở hệ PREDICT thì trả lại giá trị max+1 (không có nghĩa).
tham số	string
trả lại	int

insert

mục đích	thêm 1 đặc trưng cùng stt đi kèm của nó. Hàm này chỉ phục vụ cho việc load dữ liệu
tham số	pair<string, size_t>
trả lại	void

print

mục đích	ghi dữ liệu (đặc trưng và stt) ra file bên ngoài. địa chỉ dẫn đến file này được định nghĩa trong tham số.
tham số	string
trả lại	void

load

mục đích	đọc dữ liệu từ 1 file bên ngoài.
tham số	string
trả lại	bool

Feats

Class quản lý và lưu lại các đặc trưng và nhãn.

```
typedef pair<int, set<int>*> Feat;
```

first : nhãn (lable). Có 2 giá trị 0 và 1 tương ứng với SPACE (dấu phân tách từ) và UNDERSCORE(2 âm tiết liên tiếp là 1 từ).

second : là 1 con trỏ chỉ đến 1 set các stt của các đặc trưng (âm tiết, đặc điểm chữ hoa, chữ thường, chữ số, hay từ điển).

```
private:
    vector<Feat*> feats_;
```

Class lưu lại và thao tác trực tiếp lên feats_.

size

mục đích	kích thước của feats_
tham số	void
trả lại	int

get

mục đích	lấy ra toàn bộ feats_
tham số	void
trả lại	vector<Feat*>*

add

mục đích	thêm vào 1 Feat
tham số	Feat*
trả lại	void

type

mục đích	trả lại đặc trưng (type) của 1 âm tiết
tham số	string
trả lại	string

token

mục đích	phân đoạn 1 câu thành các âm tiết, nếu là để học máy thì chỉ cần gặp dấu SPACE hoặc UNDER là tách. Còn nếu là để tách từ, thì phải tách cả những dấu và kí hiệu.
tham số	string, FeatsReference
trả lại	vector<string>*

clear

mục đích	xóa toàn bộ dữ liệu trong feats_
tham số	void
trả lại	void

SylMap

Class quản lý và kiểm tra các âm tiết tiếng Việt.

```
private:
    map<string, string> syl_;
```

isVNESE

mục đích	kiểm tra 1 string có phải là âm tiết tiếng Việt hay không
tham số	string
trả lại	bool

size

mục đích	trả lại kích thước của syl_
tham số	void
trả lại	int

getNum

mục đích	trả lại âm tiết theo số thứ tự
tham số	int
trả lại	string

Machine

Class thực hiện việc học máy.

extract

mục đích	chuyển 1 câu thành các feats
tham số	string
trả lại	void

getProblem

mục đích	chuyển Feats thành problem
tham số	void
trả lại	problem*

delProblem

mục đích	xóa toàn bộ dữ liệu của problem
tham số	void
trả lại	void

training

mục đích	tiến hành học máy
tham số	void
trả lại	void

print

mục đích	lưu lại strmap và model ra các file dongdu.map và dongdu.model
tham số	void
trả lại	void

close_test

mục đích	tiến hành test chính bản thân dữ liệu huấn luyện
tham số	void
trả lại	double

load

mục đích	đọc dữ liệu từ file dongdu.map và dongdu.model
tham số	void
trả lại	bool

segment

mục đích	thực hiện tách từ đối với 1 câu
tham số	string
trả lại	string

FeaturesSelection

Class thực hiện việc lựa chọn đặc trưng. Những đặc trưng có weight là 0 được coi là những giá trị không cần thiết và sẽ bị loại bỏ. Với dạng L1R_LR thì lượng đặc trưng có nghĩa thường chỉ chiếm 1/40.